Command Line Development Tools

Session 901

Command Line Development Tools

Stan Shebs
Senior Engineer, Core Tools

Introduction

- Mac OS X has a full set of command-line tools
- These tools speed Mac OS X ports
- Mac OS X is a familiar place for Unix applications

Topics

- How to build and debug using command-line tools
- What tools are available on a standard system
- Unique features of Mac command-line tools
- Porting issues
- Documentation and resources

Command-Line Tools?

- On Macs!
- Familiarity: 30 years of productivity on Unix
- Power: tools are composable

```
diff old.cc new.cc | grep fo[po]_fn
```

- Leverage: thousands of tools delivered across the industry

Terminal Program

- Launch Terminal
  - **[Applications -> Utilities -> Terminal]**
  - **/Applications/Utilities/Terminal.app**
- Uses multiple windows
- Each window runs a separate shell
- Lots of preferences
- Delivered on ALL systems

Demo

Sean Fagan
Engineer

The Catechism

```
% emacs foo.c
% cat foo.c
main() { printf("hello\n"); }
% cc -g foo.c
% ./a.out
hello
% gdb a.out
(gdb) break main
Breakpoint 1 at 0x1eac: file foo.c, line 1.
(gdb) run
…  1 main() {printf("hello\n"); }
(gdb)
```

- What else do you need? :-)

Shells

- The shell interprets commands
- Different shells have different scripting languages
- tcsh
- zsh
- bash

Editor Options

- Emacs, the One True Editor
  - Highly customizable
  - 10.1 standard version is 20.7.1
  - X11 version downloadable
- Vi
  - “Because we value diversity”
  - Simple and easy to learn
- Pico, etc.

Demo

Sean Fagan
Engineer

Language Options

- C
- C++
- Objective-C, Objective-C++
- Java (1.3.1)
- Perl, tcl, PHP, Python (in 10.2)
- Third parties: Metrowerks, AbSoft
- Downloads: G77 (Fortran), GNAT (Ada), Lisp, etc.

Build Tool Options

- GNU Make
  - The standard
- BSD Make
  - For compatibility only
- Jam/pbxbuild
  - Uses Project Builder projects
  - Apple's jam extended from Net version

Other Useful Tools

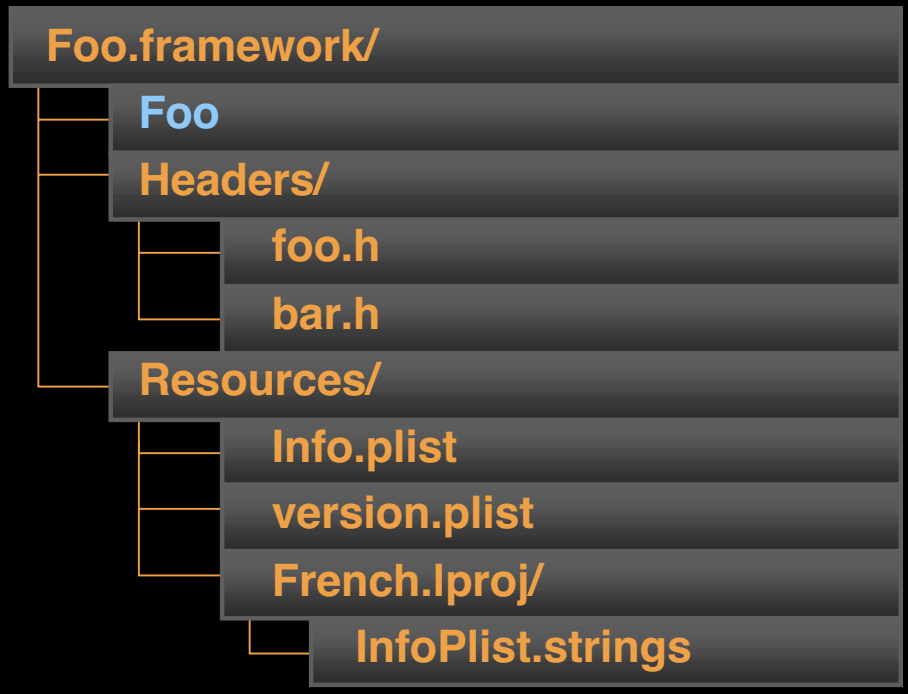
- Collection most similar to FreeBSD
  - CVS, RCS, SCCS for source code control
  - troff, nroff, eqn, pic, etc text tools
  - lex, yacc, flex, bison
  - ranlib, ar, as, otool, autoconf
- > 400 programs in /bin and /usr/bin

APIs

- Most of POSIX, but not all of it
    - Report POSIX nonconformance
- OpenGL and GLUT
- Carbon and Cocoa for GUI
- IO Kit for drivers and other kernel extensions
- Several X11 systems available

Frameworks

- A framework consists of a shared library (dylib), headers, and resources



Using Frameworks

- **#include <Foo/foo.h>** to compile
- **-framework Foo** to link
- Most in **/System/Library/Frameworks**
- **-F*path*** add place to find frameworks
  - Affects both compiler and linker
- **/usr/include** has normal Unix headers

Basic Compiler Options

- The MAC OS X compiler is GCC
- The usual GCC options work
- Many additional options and language extensions

  -faltivec enables AltiVec types and built-in functions

  -mdynamic-no-pic for apps, makes nonrelocatable but can still access relocatable dylibs, more efficient than default code gen

Building Libraries

- ar + ranlib to create statically linked libraries (libfoo.a)
- **cc -dynamiclib** for dynamically linked libraries (dylibs)
- GNU libtool (aka glibtool) usually knows how to build
  - May need to update libtool bits in source distro

Versions

- Lower layers of system called “Darwin”
  - Open source, no frameworks, no GUI
- **sw_vers** reports Mac OS X version—10.1.3, etc.
- **uname -a** reports Darwin version—5.3, etc.
- Correspondence
  - Mac OS X 10.0 == Darwin 1.3.1
  - 10.1 == 1.4
  - 10.1.[123] == 5.[123]

Basic Porting

- GNU configure/make usually “just works”
- Maybe update **config.guess** and **config.sub**:
  http://savannah.gnu.org/cgi-bin/viewcvs/config/config/
- Symlink cc to gcc and c++ to g++ for some ports
- Do not rely on predefined macros—test for features or add **-DMACOSX** in makefile

Interface Porting

- Curses (ncurses) “just works”
- Aqua-fied tcl/tk and Qt are now available
- X11 available if users run X11 server
- OpenGL and GLUT are standard
    - Must translate **#include <GL/gl.h>** into **#include <OpenGL/gl.h>**

Common Problems

- Compiler uses precompiled headers by default
  - Rarely useful for Unix ports
  - Use **-no-cpp-precomp** to suppress
- Missing declarations
- Missing functions
  - Reimplement or #ifdef sources
- Extra definitions
  - Use **-multiply_defined suppress**

Less Common Problems

- Designated initializers not available with 2.95.2-based compiler
  - **int arr1[10] = { [5] 45, [1] 92 };** (GNU)
  - **int arr2[10] = { [5] = 45, [1] = 92 };** (C99)
- **dlopen()** not directly available
  - Can emulate functionality
- Some pthread functions are missing
- Namespace conflicts
  - Use **-twolevel_namespace**
- Run ranlib after ar

Demo

Sean Fagan
Engineer

More Porting

- Add autoconf macros to test OS features
  - Assume lowest system version possible if shipping binaries
  - Read "GNU Autoconf, Automake and Libtool" (New Riders, 2000)
- **AvailabilityMacros.h** is in Jaguar

Caveats

- Do not abuse root
  - Rename tools
  - symlink headers
- Mach-O object files very different
  - No standard GNU binutils port
- Watch out for HFS+ vs. UFS issues
  - **foo.c** and **Foo.C** in same directory

Debugging With GDB

- The Mac OS X debugger is GDB
- Currently based on FSF's 5.0 release
- Many additional commands; Objective-C, Mach support, frameworks, etc.

```
future-break fmwk.c:432
print-object objc_obj
info mach-tasks
```

GCC 3

- Current compiler based on 2.95.2
- Next version of compiler will be based on 3.1
- Features
    - More optimization
    - Better C++ conformance
    - PFE for precompiled headers
- Available for testing now

Using GCC 3

- 2.95.2 always at **/usr/bin/gcc2**
- Latest version 3 always at **/usr/bin/gcc3**
- **/usr/bin/cc** and **/usr/bin/gcc** symlink to one or other of gcc[23]
- Script gcc_select, run as root, chooses
  - **gcc_select 2**
  - **gcc_select 3**
- Aux programs and headers also relinked

Using PB

- Can build legacy targets with IDE
- Many defaults and flags set up correctly for Mac OS X
- Easier access to documentation
- **pbxbuild** command available for scripts

Demo

Sean Fagan
Engineer

Technical Documentation

- Command line tools have **man** pages
  - At the terminal prompt: **man <toolname>**
- Reference docs in **/Developer/Documentation/DevTools** for
  - Compiler
  - Preprocessor
  - gdb
  - MachORuntime

For More Information

- ADC Tools pages
  http://developer.apple.com/tools/
- Darwin pages
  http://www.apple.com/darwin/
- VersionTracker
  http://www.versiontracker.com
- GNU-Darwin at SourceForge
  http://www.gnu-darwin.sf.net
- Fink at SourceForge
  http://www.fink.sf.net

Roadmap

105 Porting UNIX Apps to Mac OS X: Helpful tips to speed porting your apps	Civic **Tue., 5:00pm**
FF000 Technical Documentation: Give feedback to the documentation team	Room J1 **Wed., 10:30am**
903 Exploring the Project Builder IDE: Learn all about Project Builder's features	Hall 2 **Wed., 5:00pm**
905 Apple Performance Tools: See the tools supplied to optimize your app	Hall 2 **Thurs., 5:00pm**

Who to Contact

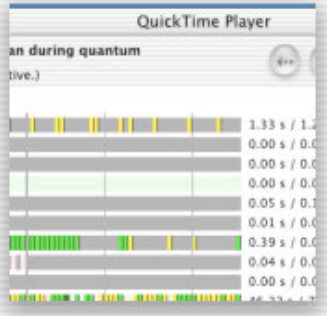
Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

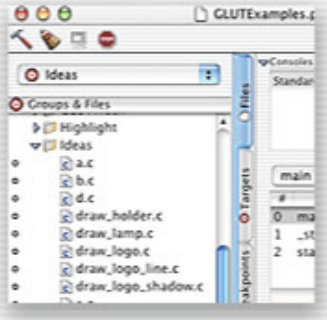
Mailing Lists at Apple
http://lists.apple.com
- darwin
- projectbuilder-users

Development Tools Engineering Feedback
macosx-tools-feedback@group.apple.com

http://developer.apple.com/wwdc2002/urls.html

Q&A





Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002